

## NAG Library Routine Document

### F08TEF (DSPGST)

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

#### 1 Purpose

F08TEF (DSPGST) reduces a real symmetric-definite generalized eigenproblem  $Az = \lambda Bz$ ,  $ABz = \lambda z$  or  $BAz = \lambda z$  to the standard form  $Cy = \lambda y$ , where  $A$  is a real symmetric matrix and  $B$  has been factorized by F07GDF (DPPTRF), using packed storage.

#### 2 Specification

```
SUBROUTINE F08TEF ( ITYPE, UPLO, N, AP, BP, INFO )
  INTEGER           ITYPE, N, INFO
  REAL (KIND=nag_wp) AP(*), BP(*)
  CHARACTER(1)     UPLO
```

The routine may be called by its LAPACK name *dspgst*.

#### 3 Description

To reduce the real symmetric-definite generalized eigenproblem  $Az = \lambda Bz$ ,  $ABz = \lambda z$  or  $BAz = \lambda z$  to the standard form  $Cy = \lambda y$  using packed storage, F08TEF (DSPGST) must be preceded by a call to F07GDF (DPPTRF) which computes the Cholesky factorization of  $B$ ;  $B$  must be positive definite.

The different problem types are specified by the argument ITYPE, as indicated in the table below. The table shows how  $C$  is computed by the routine, and also how the eigenvectors  $z$  of the original problem can be recovered from the eigenvectors of the standard form.

ITYPE	Problem	UPLO	$B$	$C$	$z$
1	$Az = \lambda Bz$	'U' 'L'	$U^T U$ $LL^T$	$U^{-T} A U^{-1}$ $L^{-1} A L^{-T}$	$U^{-1} y$ $L^{-T} y$
2	$ABz = \lambda z$	'U' 'L'	$U^T U$ $LL^T$	$U A U^T$ $L^T A L$	$U^{-1} y$ $L^{-T} y$
3	$BAz = \lambda z$	'U' 'L'	$U^T U$ $LL^T$	$U A U^T$ $L^T A L$	$U^T y$ $L y$

#### 4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

## 5 Arguments

- 1: ITYPE – INTEGER *Input*  
*On entry:* indicates how the standard form is computed.  
 ITYPE = 1  
     if UPLO = 'U',  $C = U^{-T}AU^{-1}$ ;  
     if UPLO = 'L',  $C = L^{-1}AL^{-T}$ .  
 ITYPE = 2 or 3  
     if UPLO = 'U',  $C = UAU^T$ ;  
     if UPLO = 'L',  $C = L^TAL$ .  
*Constraint:* ITYPE = 1, 2 or 3.
- 2: UPLO – CHARACTER(1) *Input*  
*On entry:* indicates whether the upper or lower triangular part of  $A$  is stored and how  $B$  has been factorized.  
 UPLO = 'U'  
     The upper triangular part of  $A$  is stored and  $B = U^T U$ .  
 UPLO = 'L'  
     The lower triangular part of  $A$  is stored and  $B = LL^T$ .  
*Constraint:* UPLO = 'U' or 'L'.
- 3: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrices  $A$  and  $B$ .  
*Constraint:*  $N \geq 0$ .
- 4: AP(\*) – REAL (KIND=nag\_wp) array *Input/Output*  
**Note:** the dimension of the array AP must be at least  $\max(1, N \times (N + 1)/2)$ .  
*On entry:* the upper or lower triangle of the  $n$  by  $n$  symmetric matrix  $A$ , packed by columns.  
 More precisely,  
     if UPLO = 'U', the upper triangle of  $A$  must be stored with element  $A_{ij}$  in  $AP(i + j(j - 1)/2)$  for  $i \leq j$ ;  
     if UPLO = 'L', the lower triangle of  $A$  must be stored with element  $A_{ij}$  in  $AP(i + (2n - j)(j - 1)/2)$  for  $i \geq j$ .  
*On exit:* the upper or lower triangle of AP is overwritten by the corresponding upper or lower triangle of  $C$  as specified by ITYPE and UPLO, using the same packed storage format as described above.
- 5: BP(\*) – REAL (KIND=nag\_wp) array *Input*  
**Note:** the dimension of the array BP must be at least  $\max(1, N \times (N + 1)/2)$ .  
*On entry:* the Cholesky factor of  $B$  as specified by UPLO and returned by F07GDF (DPPTRF).
- 6: INFO – INTEGER *Output*  
*On exit:* INFO = 0 unless the routine detects an error (see Section 6).

## 6 Error Indicators and Warnings

INFO < 0

If INFO =  $-i$ , argument  $i$  had an illegal value. An explanatory message is output, and execution of the program is terminated.

## 7 Accuracy

Forming the reduced matrix  $C$  is a stable procedure. However it involves implicit multiplication by  $B^{-1}$  if (ITYPE = 1) or  $B$  (if ITYPE = 2 or 3). When F08TEF (DSPGST) is used as a step in the computation of eigenvalues and eigenvectors of the original problem, there may be a significant loss of accuracy if  $B$  is ill-conditioned with respect to inversion. See the document for F08SAF (DSYGV) for further details.

## 8 Parallelism and Performance

F08TEF (DSPGST) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

## 9 Further Comments

The total number of floating-point operations is approximately  $n^3$ .

The complex analogue of this routine is F08TSF (ZHPGST).

## 10 Example

This example computes all the eigenvalues of  $Az = \lambda Bz$ , where

$$A = \begin{pmatrix} 0.24 & 0.39 & 0.42 & -0.16 \\ 0.39 & -0.11 & 0.79 & 0.63 \\ 0.42 & 0.79 & -0.25 & 0.48 \\ -0.16 & 0.63 & 0.48 & -0.03 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.09 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.09 & 0.34 & 1.18 \end{pmatrix},$$

using packed storage. Here  $B$  is symmetric positive definite and must first be factorized by F07GDF (DPPTRF). The program calls F08TEF (DSPGST) to reduce the problem to the standard form  $Cy = \lambda y$ ; then F08GEF (DSPTRD) to reduce  $C$  to tridiagonal form, and F08JFF (DSTERF) to compute the eigenvalues.

### 10.1 Program Text

```

Program f08tefe

!      F08TEF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
      Use nag_library, Only: dpptrf, dspgst, dsptrd, dsterf, nag_wp
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
      Integer                     :: i, info, j, n
      Character (1)               :: uplo

```

```

! .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: ap(:), bp(:), d(:), e(:), tau(:)
! .. Executable Statements ..
Write (nout,*) 'F08TEF Example Program Results'
! Skip heading in data file
Read (nin,*)
Read (nin,*) n

Allocate (ap(n*(n+1)/2),bp(n*(n+1)/2),d(n),e(n-1),tau(n))

! Read A and B from data file

Read (nin,*) uplo
If (uplo=='U') Then
  Read (nin,*)((ap(i+j*(j-1)/2),j=i,n),i=1,n)
  Read (nin,*)((bp(i+j*(j-1)/2),j=i,n),i=1,n)
Else If (uplo=='L') Then
  Read (nin,*)((ap(i+(2*n-j)*(j-1)/2),j=1,i),i=1,n)
  Read (nin,*)((bp(i+(2*n-j)*(j-1)/2),j=1,i),i=1,n)
End If

! Compute the Cholesky factorization of B
! The NAG name equivalent of dpptrf is f07gdf
Call dpptrf(uplo,n,bp,info)

Write (nout,*)
If (info>0) Then
  Write (nout,*) 'B is not positive definite.'
Else

! Reduce the problem to standard form C*y = lambda*y, storing
! the result in A
! The NAG name equivalent of dspgst is f08tef
Call dspgst(1,uplo,n,ap,bp,info)

! Reduce C to tridiagonal form T = (Q**T)*C*Q
! The NAG name equivalent of dsptrd is f08gef
Call dsptrd(uplo,n,ap,d,e,tau,info)

! Calculate the eigenvalues of T (same as C)
! The NAG name equivalent of dsterf is f08jff
Call dsterf(n,d,e,info)

If (info>0) Then
  Write (nout,*) 'Failure to converge.'
Else

! Print eigenvalues

  Write (nout,*) 'Eigenvalues'
  Write (nout,99999) d(1:n)
  End If
End If

99999 Format (3X,(9F8.4))
End Program f08tefe

```

## 10.2 Program Data

```

F08TEF Example Program Data
  4                               :Value of N
  'L'                             :Value of UPLO
  0.24
  0.39 -0.11
  0.42  0.79 -0.25
-0.16  0.63  0.48 -0.03      :End of matrix A
  4.16
-3.12  5.03
  0.56 -0.83  0.76
-0.10  1.09  0.34  1.18      :End of matrix B

```

### **10.3 Program Results**

F08TEF Example Program Results

Eigenvalues

-2.2254 -0.4548 0.1001 1.1270

---